Data Acquisition

ENEE4304

May 2016

Need For Data Acquisition

- There are many applications where it is necessary to know, simultaneously, the measured values of several variables associated with a particular process, machine or situation.
- Examples are measurements of flow rates, levels, pressures and compositions in a distillation column, temperature measurements at different points in a nuclear reactor core, and components of velocity and acceleration for an aircraft.

- It would be extremely uneconomic to have several completely independent systems,and a single multi-input/multi-output **data acquisition system** is used.
- Here several elements are ‘time shared’ amongst the different measured variable inputs.
- This is the technique of **time division multiplexing**

- The oil, water and gas industries are characterised by complex distribution systems involving the transfer of fluids by long pipelines from producing to consuming areas.
- Similarly, an electricity distribution system involves the transfer of electrical power from power stations to consumers, via a network of high voltage cables.
- These systems also include several items of equipment or plant, e.g. pumping stations, compressors, storage tanks and transformers, each with associated measured variables.
- These plant items are often located several miles from each other, in remote areas. It is essential for the effective supervision of these distribution systems that all relevant network measurement data are transmitted to a central control point.

- To do this a complex **communications system** is required. This usually consists of a **master station** (at the central control point) and several **outstations** (at the plant items).
- The system must be capable of transmitting large amounts of information in two directions (M/S to O/S and O/S to M/S), over long distances, in the presence of interference and noise.
- The principles of **parallel digital signalling**, **serial digital signalling**, **error detection/correction** and **frequency shift keying**, which are used in communications systems, and concludes by describing the implementation of communications systems for measurement data with special regard to the **Fieldbus standard**.

Data acquisition hardware

- Data acquisition hardware is available in many forms.
- **Plug-in expansion bus boards**, which are plugged directly into the computer's expansion bus, are a commonly utilized item of DAQ hardware.
- **Intelligent stand-alone loggers and controllers**, which can be monitored, controlled and configured from the computer via an RS-232 interface, and yet can be left to operate independently of the computer.
- **Remote stand-alone instrument that can be configured and controlled by the computer, via the IEEE-488 communication interface**.

Data acquisition software **(for info only)**

- **Data acquisition hardware does not work without software, because it is the software running on the computer that transforms the system into a complete data acquisition, analysis, display, and control system.**
- Application software runs on the computer under an operating system that may be single-tasking (like DOS) or multitasking (like Windows, Unix, OS2), allowing more than one application to run simultaneously.
- The application software can be:
  - a full screen interactive panel,
  - a dedicated input/output control program,
  - a data logger,
  - a communications handler, or a combination of all of these.

(for info only)

Software Options for Hardware Programming

1. **Program the registers of the data acquisition hardware directly.**
2. **Utilize low-level driver software**, usually provided with the hardware, **to develop a software application for the specific tasks required.**
3. **Utilize off-the-shelf application software** – this can be application software, provided with the hardware itself, which performs all the tasks required for a particular application.
4. **Use third party packages such as LabVIEW** and **Labtech Notebook** which provide a graphical interface for programming the tasks required of a particular item of hardware, as well as providing tools to analyze and display the data acquired.

Host computer

- The PC used in a data acquisition system can greatly affect the speeds at which data can be continuously and accurately acquired, processed, and stored for a particular application.
- **Depending on the particular application, The factors that can all have an impact on the speed at which the computer is able to continuously acquire data**

1) The microprocessor speed,

2) hard disk access time,

3) disk capacity and

4) the types of data transfer available,

Real-time processing

- **If real-time processing of the acquired data is needed, the performance of the computer's processor is paramount. A minimum requirement for high frequency signals acquired at high sampling rates would be a 32-bit processor with its accompanying coprocessor, or alternatively a dedicated plug-in processor**.
- **Low frequency signals**, for which only a few samples are processed each second, would obviously not require the same level of processing power. A low-end PC would therefore be satisfactory.
- Clearly, the performance requirements of the host computer must be matched to the specific application.
- As with all aspects of a data acquisition system the choice of computer is a compromise between cost and the current and future requirements it must meet.

Data Loggers (for info only)

- A **data logger** (also **datalogger** or **Data recorder**) is an electronic device that records data over time or in relation to location either with a built in instrument or sensor or via external instruments and sensors.
- Data loggers are based on a digital processor (or computer).
- They generally are small, battery powered, portable, and equipped with a microprocessor, internal memory for data storage, and sensors.
- Some data loggers interface with a personal computer and utilize software to activate the data logger and view and analyze the collected data, while others have a local interface device (keypad, LCD) and can be used as a stand-alone device.

Computer plug-in I/O (for info only)

- **Plug-in I/O boards are plugged directly into the computers expansion bus, are generally compact, and also represent the fastest method of acquiring data to the computers memory and/or changing outputs**.
- Along with these advantages, plug-in boards often represent the lowest cost alternative for a complete data acquisition and control system and are therefore a commonly utilized item of DAQ hardware

Computer plug-in I/O (for info only)

- Examples of plug-in I/O boards are:

1) multiple analog input A/D boards,

2) multiple analog output D/A boards,

3) digital I/O boards,

4) counter/timer boards,

5) specialized controller boards (such as stepper/servo motor controllers)

6) Specialized instrumentation boards (such as digital oscilloscopes).

Distributed I/O (for info only)

- **Often sensors must be remotely located from the computer in which the processing and storage of the data takes place.**
- This is especially true in industrial environments where sensors and actuators can be located in hostile environments over a wide area, possibly hundreds of meters away.
- In noisy environments, it is very difficult for very small signals received from sensors such as thermocouples and strain gauges (in the order of mV) to survive transmission over such long distances, especially in their raw form, without the quality of the sensor data being compromised.

(for info only)

- An alternative to running long and possibly expensive sensor wires, is the use of distributed I/O, which is available in the form of **signal conditioning modules remotely located** near the sensors to which they are interfaced.
- One module is required for each sensor used, allowing for high levels of modularity (single point to hundreds of points per location). While this can add reasonable expense to systems with large point counts, the benefits in terms of signal quality and accuracy may be worth it.

Distributed I/O → Digital Transmitter

- One of the most commonly implemented forms of distributed I/O is the digital transmitter. These intelligent devices perform all required signal conditioning functions and contain a micro-controller and A/D converter, to perform the digital conversion of the signal within the module itself.
- Converted data is transmitted to the computer via an RS-232 or RS-485 communications interface.
- The use of RS-485 multi-drop networks, as shown in Figure 1.3, reduces the amount of cabling required, since each signal-conditioning module shares the same cable pair. Linking up to 32 modules, communicating over long distances, is possible when using the RS-485 multi-drop network.
- However, since very few computers have built in support for the RS-485 standard, an RS-232 to RS-485 converter is required to allow communications between the computer and the remote modules.

(for info only)

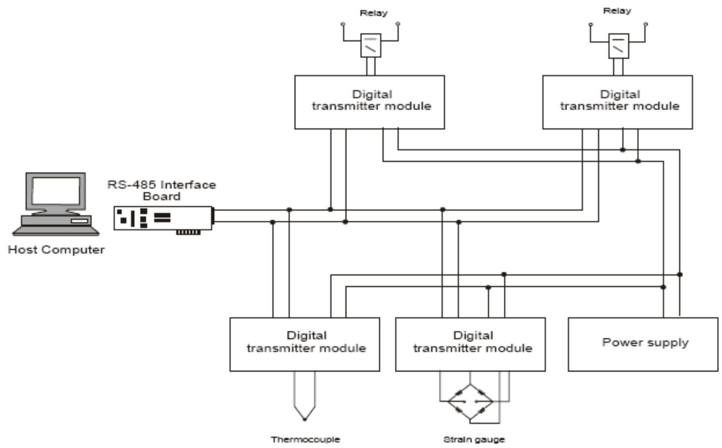


Figure 1.3
Distributed I/O – digital transmitter modules

Instrumentation

(for info only)

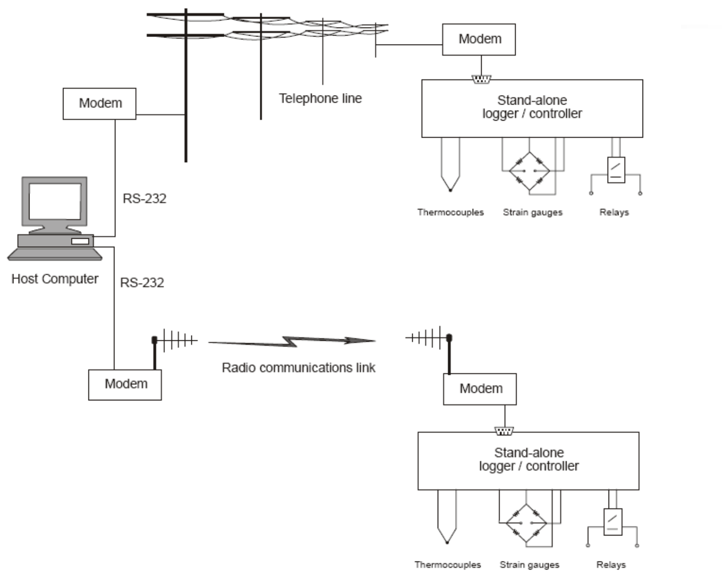


Figure 1.6
Remote connection to a stand-alone logger/controller via a telephone or radio communications network

(for info only)

- Where an application requires more than one logger/controller, each unit is connected within an RS-485 multi-drop network. A signal unit, deemed to be the host unit, can be connected directly to the host computer via the RS-232 serial interface, as shown in Figure 1.7, thus avoiding any requirement for an RS-232 to RS-485 serial interface card.
- The same methods of programming or logging data from each logger/controller are available either via the serial communications network or via using portable and reusable memory cards.

(for info only)

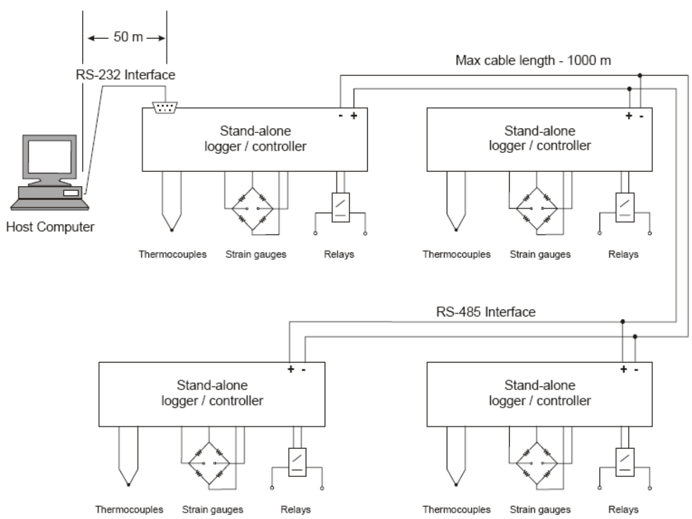


Figure 1.7
Distributed logger/controller network

1.3.4 IEEE-488 (GPIB) remote programmable instruments

- The communications standard now known as GPIB (General Purpose Interface Bus), Was originally developed by Hewlett-Packard in 1965 as a digital interface for interconnecting and controlling their programmable test instruments.
- Originally referred to as the Hewlett Packard Interface Bus (HPIB), its speed, flexibility and usefulness in connecting instruments in a laboratory environment led to its widespread acceptance, and finally to its adoption as a world standard (IEEE-488).
- Since then, it has undergone improvements (IEEE-488.2) and SCPI (Standard Commands for Programmable Instruments), to standardize how instruments and their controllers communicate and operate.
- Evolving from the need to collect data from a number of different stand-alone instruments in a laboratory environment, **the GPIB is a high-speed parallel communications interface that allows the simultaneous connection of up to 15 devices or instruments on a short common parallel data communications bus**.
- **Devices must be placed within 3 meters or so of the host controller/ computer**

- A device connected to the bus can send data (bytes) to 14 other devices on the bus
- GPIB allows data to be sent at whatever rate the devices on the bus operate.
- Hardware consideration limit the max speed of data transmission to 250 kbytes/s (= 2 Mbits/s).
- GPIB is used to communicate with a set of instruments with the same interface for setting an automatic measurement and control system by a network of instruments

(for info only)

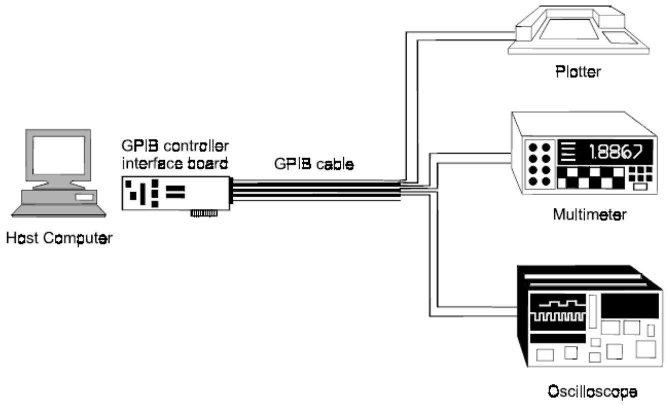
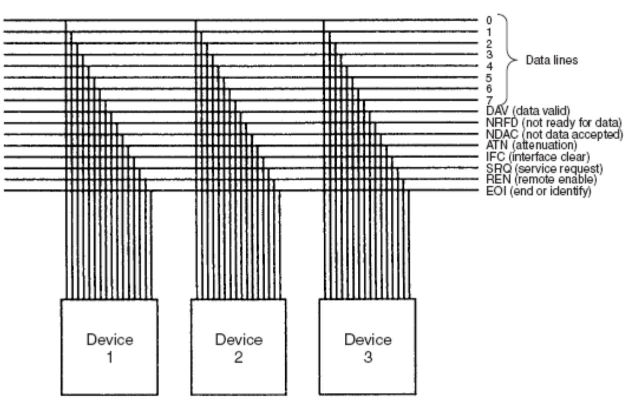


Figure 1.8
A typical GPIB system configuration

Instrumentation

GPIB Bus Structure



-16 lines

-8 carry data (or commands)

-8 are bus control lines: 3 provide handshaking to coordinate data transfer and 5 are other bus control functions

Instrumentation

Device on GPIB

Are classified into:

- Listener: may receive data over the bus
- Talker: may send data over the bus
- Listener/ Talker:
- Controller: at least one device must act as a controller (usually PC that also act as listener and talker)

*Any one device can perform combination of these functions.

Only 15 devices are allowed on the bus, if more are used, this causes overloading and unreliable operation

- GPIB signal logic:1) TTL voltage levels
  2) negative logic ➔ < 0.8V (logic 1); > 2.5 (logic 0)
- Lines are driven by open collector drivers

(for info only)

- 3 Handshaking Lines

• DAV (Data valid)	This goes to a logic zero when the data on the eight data lines is valid.
• NFRD (Not ready for data)	This goes to logic zero when the receiver unit is ready to accept data.
• NDAC (Not data accepted)	This goes to logic zero when the receiver unit has finished receiving data.

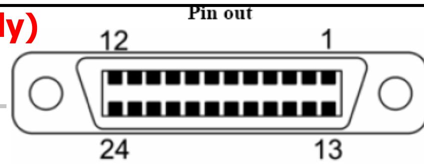
- 5 Control Lines

• ATN (Attention)	This is a general control signal that is used for various purposes to control the use of data lines and specify the send and receive devices to be used.
• IFC (Interface clear)	The controller uses this status line to put the interface into a wait state.
• SRQ (Service request)	This is an interrupt status line that allows high priority devices such as alarms to interrupt current bus traffic and get immediate access to the bus.
• REN (Remote enable)	This status line is used to specify which of two alternative sets of device programming data are to be used.
• EOI (End of output or identify)	This status line is used by the sending unit to indicate that it has finished transmitting data.

(for info only)

IEEE-488 connector / FYI

Pin out



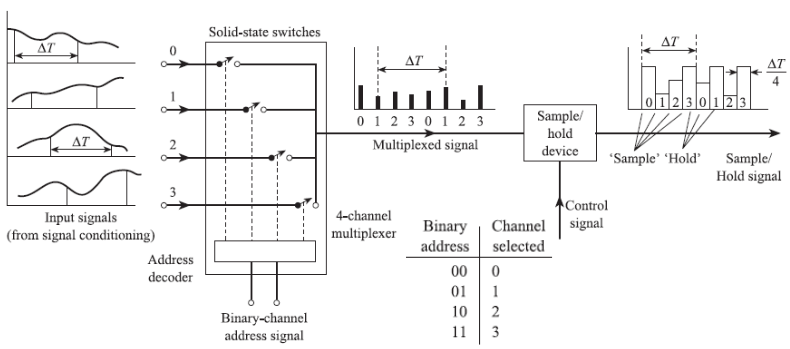
A female IEEE-488 connector

Pin 1	DIO1	Data input/output bit.
Pin 2	DIO2	Data input/output bit.
Pin 3	DIO3	Data input/output bit.
Pin 4	DIO4	Data input/output bit.
Pin 5	EOI	End-or-identify.
Pin 6	DAV	Data valid.
Pin 7	NRFD	Not ready for data.
Pin 8	NDAC	Not data accepted.
Pin 9	IFC	Interface clear.
Pin 10	SRQ	Service request.
Pin 11	ATN	Attention.
Pin 12	SHIELD	
Pin 13	DIO5	Data input/output bit.

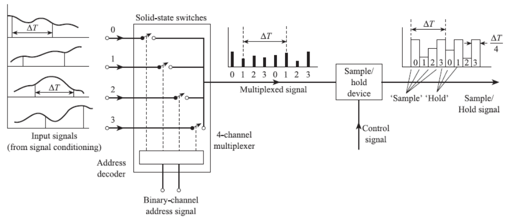
Pin 15	DIO7	Data input/output bit.
Pin 16	DIO8	Data input/output bit.
Pin 17	REN	Remote enable.
Pin 18	GND	(wire twisted with DAV)
Pin 19	GND	(wire twisted with NRFD)
Pin 20	GND	(wire twisted with NDAC)
Pin 21	GND	(wire twisted with IFC)
Pin 22	GND	(wire twisted with SRQ)
Pin 23	GND	(wire twisted with ATN)
Pin 24	Logic ground	

Instrumentation

Time Division Multiplexing



- The multiplexer output signal is thus a series of samples taken from different measurement signals at different times.
- In **sequential** addressing the channels are addressed in order, i.e. first 0, followed by 1, then 2 and 3, returning to channel 0 and repeating, so that the pattern of samples for the multiplexed signal is as shown in the diagram
- **Random** addressing, whereby an observer selects a channel of interest at random, is also possible.



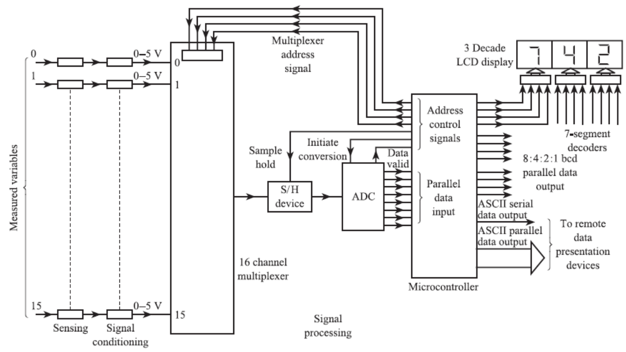
- If ΔT is the sampling interval, i.e. the time interval between samples of a given input, e.g. 0 or 1, then the corresponding sampling frequency $f_S = 1/\Delta T$ must satisfy the conditions for the Nyquist sampling theorem
- These require that f_S be greater than or equal to $2f_{MAX}$, where f_{MAX} is the highest significant frequency present in the power spectral density of the measurement signal.
- Here, Four samples occur during ΔT, so that the number of samples per second for the *multiplexed* signal is $4f_S$. In general, for m signals, each sampled f_S times per second, the **number of samples per second for the multiplexed signal is:**

$$f_S^M = mf_S$$

- Different measured variables may have frequency spectra with different maximum frequencies: thus the power spectrum of a flow measurement may extend up to 1 Hz, but that of a temperature measurement only up to 0.01 Hz.
- The sampling frequency of the flow measurement must therefore be 100 times that of the temperature measurement.
- In the multiplexed signal there will be 100 samples of the flow measurement between each temperature sample.
- The multiplexed signal is normally fed to a sampleand-hold device.

microcontroller-based data acquisition system

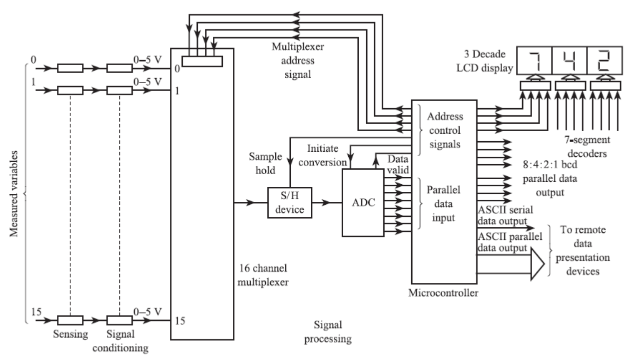
- a typical microcontroller-based data acquisition system is shown.
- The signal conditioning elements are necessary to convert sensor outputs to a common signal range, typically 0 to 5 V;
- The voltage signals are input to a 16- channel time division multiplexer, and the multiplexed signal is passed to a single sample/hold device and analogue-to-digital converter



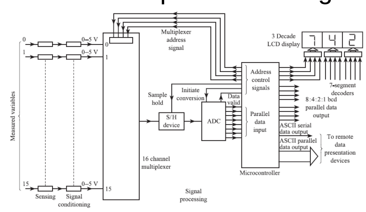
Instrumentation

microcontroller-based data acquisition system

- In cases where all the sensors are of an identical type, for example 16 thermocouples, it is more economical to multiplex the sensor output signals.
- Here the multiplexed sensor signal is input to a single signal conditioning element, such as the reference junction circuit and instrumentation amplifier, before passing to the sample/hold and ADC.

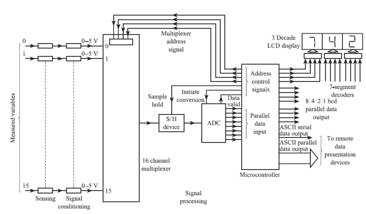


- The ADC gives a parallel digital output signal which passes to one of the parallel input interfaces of the microcontroller.
- Another parallel input/output (I/O) interface provides the address and control signals necessary for the control of multiplexer, sample/hold and ADC.
- These are a four-bit multiplexer address signal, a **sample/ hold** control signal, an **initiate conversion** signal to the ADC, and a **data valid** signal from the ADC.
- The microcontroller performs whatever calculations (on the input data) are necessary to establish the measured value of the variable.
- A common example is the solution of the non-linear equation relating thermocouple e.m.f. and temperature



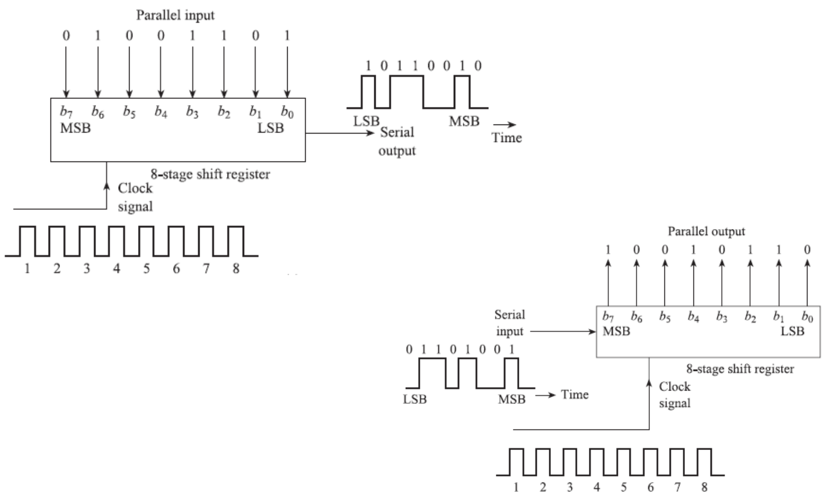
Instrumentation

- The computer converts the measured value from hexadecimal into binary coded decimal form .
- This b.c.d. data is written into a computer parallel output interface.
- Each decade is then separately converted into seven-segment code and presented to the observer using a seven-segment LCD display
- The computer also converts each decade of the b.c.d. to ASCII form
- The resulting ASCII code is then written into a serial and/or parallel output interface.
- These can transmit ASCII data in serial and/or parallel form to remote data representation devices such as a monitor, printer or host computer.



Instrumentation

Parallel & Serial Signals



Other Interfaces

- GPIB used for managing large measurement systems
- Serial Interface is often used when a single instrument is to be connected to a PC over long distance
- RS232 was originally developed in 1960s and it is slow, not flexible and rarely used on instruments, however it is used for specific applications such as reading in data from remote dc sensors and sending data to loggers
- Other more modern, serial asynchronous data transmission protocols include RS422, RS423, RS449, RS485 and USB
- RS stands for recommended standard

RS232 basic characteristics

- 1 driver, 1 receivers
- ~ 50 Feet
- Rate : 20Kb/s
- Single Ended

RS-422

- 1 driver, 10 receivers
- 4000 Feet
- Rate : 100Kb/s -10Mb/s
- Differential

RS-485

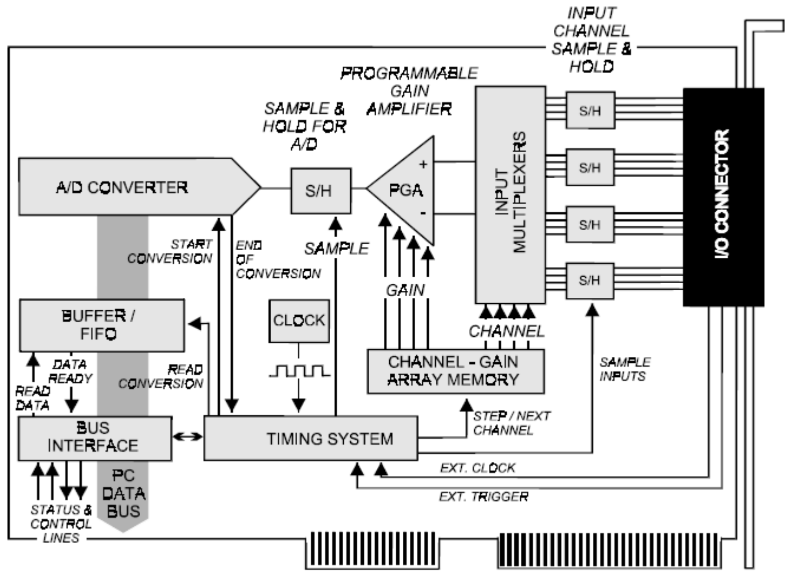
- 32 driver, 32 receivers
- 4000 Feet
- Rate : 100Kb/s -10Mb/s
- Differential

Instrumentation

(for info only)

SPECIFICATIONS		RS232	RS423	RS422	RS485
Mode of Operation		SINGLE -ENDED	SINGLE -ENDED	DIFFERENTIAL	DIFFERENTIAL
Total Number of Drivers and Receivers on One Line (One driver active at a time for RS485 networks)		1 DRIVER 1 RECVR	1 DRIVER 10 RECVR	1 DRIVER 10 RECVR	32 DRIVER 32 RECVR
Maximum Cable Length		50 FT.	4000 FT.	4000 FT.	4000 FT.
Maximum Data Rate (40ft. - 4000ft. for RS422/RS485)		20kb/s	100kb/s	10Mb/s-100Kb/s	10Mb/s-100Kb/s
Maximum Driver Output Voltage		+/-25V	+/-6V	-0.25V to +6V	-7V to +12V
Driver Output Signal Level (Loaded Min.)	Loaded	+/-5V to +/-15V	+/-3.6V	+/-2.0V	+/-1.5V
Driver Output Signal Level (Unloaded Max)	Unloaded	+/-25V	+/-6V	+/-6V	+/-6V
Driver Load Impedance (Ohms)		3k to 7k	>=450	100	54
Max. Driver Current in High Z State	Power On	N/A	N/A	N/A	+/-100uA
Max. Driver Current in High Z State	Power Off	+/-6mA @ +/-2v	+/-100uA	+/-100uA	+/-100uA
Slew Rate (Max.)		30V/uS	Adjustable	N/A	N/A
Receiver Input Voltage Range		+/-15V	+/-12V	-10V to +10V	-7V to +12V
Receiver Input Sensitivity		+/-3V	+/-200mV	+/-200mV	+/-200mV
Receiver Input Resistance (Ohms), (1 Standard Load for RS485)		3k to 7k	4k min.	4k min.	>=12k

A/D Board **(with simultaneous S&H)**



Sampling Techniques

- These techniques are discussed in the following sections:
- **Continuous channel scanning**
- **Simultaneous sampling**
- **Block mode operations**

Continuous channel scanning

- The method of sampling that facilitates the connecting of the required input channel to the A/D converter at a constant rate is known as continuous channel scanning.
- Continuous channel scanning allows channels to be sampled in a pre-determined and arbitrary order (e.g. channel 5, channel 1, channel 11), as well as at different sampling rates.
- An example of this would be the sampling of three channels in the following order (channel 5, channel 1, channel 11, channel 1). Channel 1 is being sampled at twice the rate as channels 5 & 11, which for an A/D board with throughput of 100 kHz represents a sampling rate of 50 kHz.
- Channels 5 & 11 are sampled at 25 kHz.
- There are two methods of continuous channel scanning, either under software control or by on-board hardware control using Channel Gain Array.

- When the input multiplexer switches between channels, a time skew/delay is generated between each channel sampled.
- On an A/D board being sampled at its maximum total throughput of 200 kHz, the minimum channel-to-channel time skew/delay between samples on different channels is 5 μs. Since the skew is additive from channel to channel, the total time skew between the first and last samples, when 16 channels are being sampled, is 80 μs.
- Time skew between signal measurements taken on different channels can lead to an inaccurate portrayal of the events that generated the signals as demonstrated on next slide.

Error due to continuous scanning

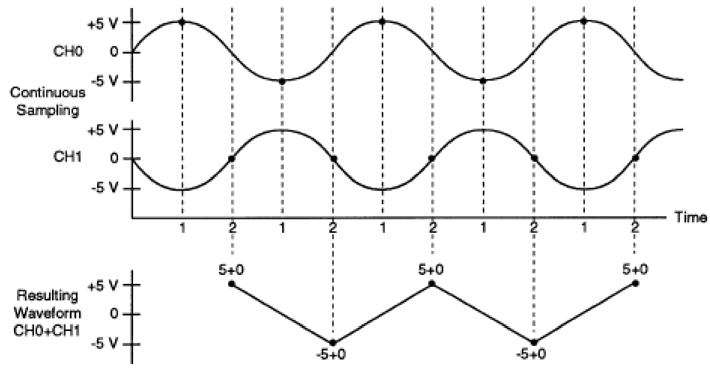


FIGURE 96.9 If the channel skew is large compared with the signal, then erroneous conclusions may result.

Simultaneous Sampling

- **Where the time relationship between each channel sampled is unimportant, or the skew/delay is negligible compared to the speed of the channel scan rate, such delays are not significant.**
- **In many applications, however, such as those dealing with accurate phase measurements or high-speed transient analysis, time skew between channels is unacceptable, since it is crucial to determine the output of several signals on different channels, at precisely the same time.**
- To avoid the timing errors introduced when continuously sampling from one input channel to the next, special applications require A/D boards capable of simultaneous sampling.
- These A/D boards are fitted with so-called simultaneous sample and hold devices on all input channels. The sample and hold device on each input channel holds the sampled data until the A/D converter can scan each channel.

Block mode triggering/Sampling / **(for info only)**

- Block mode triggering initiates an A/D conversion on all the required input channels at the maximum sampling rate of the A/D board, every time a sample trigger pulse occurs.
- A second counter is used to trigger the sampling of each of the channels at the maximum sampling rate. The number of samples to be taken in each block is typically stored by software in an on-board buffer, while the channel and gain for each sample in the block is read from the channel/gain array. The scan sequence is repeated at the next sample trigger pulse.
- Consider an example where four channels are being sampled at a total throughput rate of 20 kHz, corresponding to a channel scan rate of 5 kHz. Figure 5.23 shows that in continuous scanning mode, the total scan time is 200 μs, with the samples evenly spaced every 50 μs.
- In block trigger mode, the four samples are taken in a single scan sequence at the maximum throughput of the board. Assuming the board is capable of taking samples at 200 kHz, the time between each of these four samples is 5 μs, while the total time taken for all the samples is 20 μs instead of 200 μs.

FYI

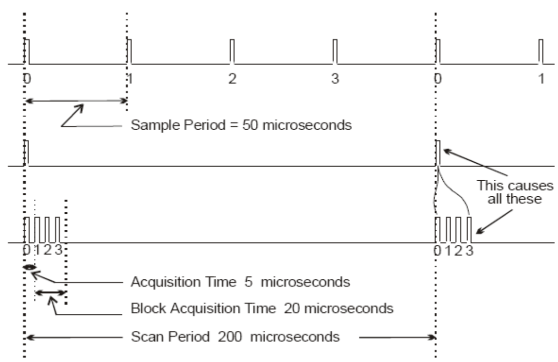


Figure 5.23
Conventional and burst trigger scanning

FYI

- Where the sampling rate remains the same, that is, a sample trigger occurs every 50 μs, the throughput of the board is increased by the number of samples taken in each sample block.
- In this case, the throughput would be increased to 80 kHz.

Important

- Maximum throughput per channel = Total throughput / # of channels
- For example if you wish to sample 4 channels at 50 kHz each, you need a board with throughput of 200 kHz

General Purpose DAQ/DAS

- Example is the AT-MIO-16

(for info only)

AT-MIO-16 (Just for reference)

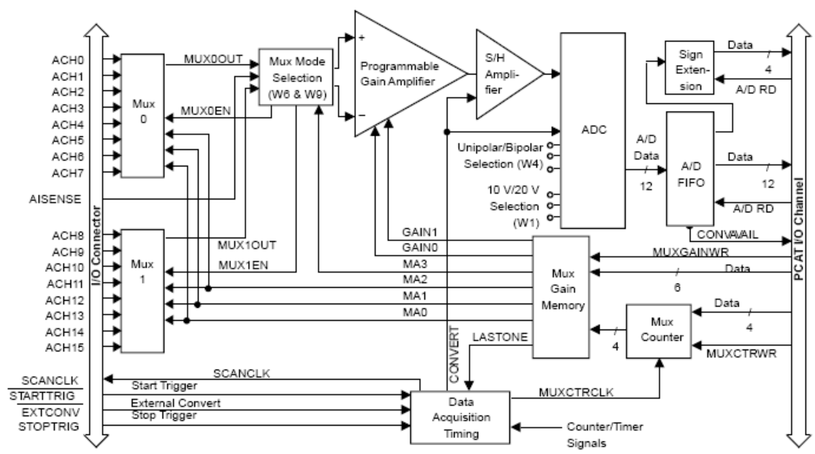
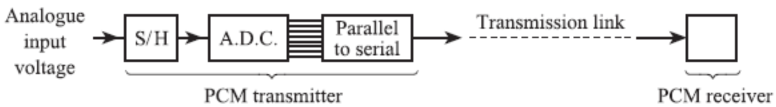


Figure 2-3. Analog Input and Data Acquisition Circuitry Block Diagram

Transmission bandwidth

- Figure below shows a simple PCM transmission system. A transmitter, consisting of a sample/hold device, ADC and parallel-to-serial converter, converts an input analogue voltage into a serial digital signal, which is sent over a transmission link to a receiver.

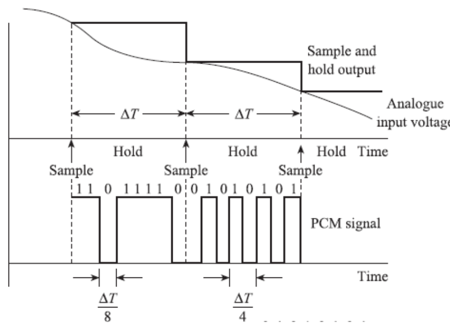


- The transmission link may be cable, radio link or optical fibre.
- In order to estimate the bandwidth required for the transmission link it is necessary to find the extent of the frequency spectrum of the PCM signal.
- We first need to find the **bit rate** of the PCM signal; this is the number of bits per second or **baud** (1 baud = 1 bit /s).

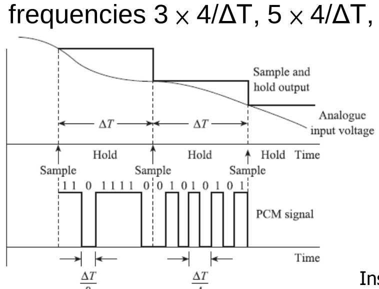
- Consider a single signal, sampled fs times per second, each sample being encoded into an *n*-bit code.
- There are fs samples per second and *n* bits per sample, so that the bit rate is: $R = nf_S$
- For *m* multiplexed signals, each sampled fs times per second, there are mfs samples per second, so that in this case the bit rate is:

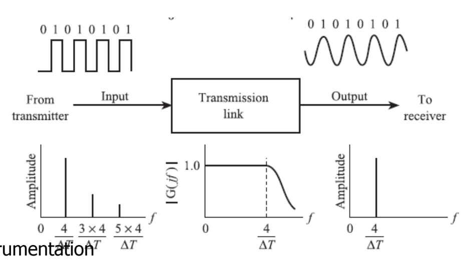
  Bit rate for m multiplexed signals $R = nmf_S$

- Figure shows corresponding time variations in input analogue voltage, sample and-hold output signal and PCM signal.
- The graphs assume that the sample/hold device is in the SAMPLE state for an infinitely short time.
- This means that, for a single signal, the time in the HOLD state is equal to the sampling interval ΔT.



- If ADC has an 8-bit encoder, i.e. $n = 8$, then eight bits (either 0 or 1) of information must be transmitted during this time interval ΔT. Thus the width of each bit of information in the PCM signal is $\Delta T/8$.
- There are 256 possible pulse patterns during each sampling interval, but the pulse pattern corresponding to 01010101 has the shortest period and the highest frequency components.
- From Figure we see that this pulse pattern is a square wave of period $\Delta T/4$. The frequency spectrum of this square wave consists of a fundamental of frequency $4/\Delta T$ Hz, together with harmonics at frequencies $3 \times 4/\Delta T$, $5 \times 4/\Delta T$,





Instrumentation

- If this square wave signal is transmitted over a link with bandwidth between 0 and a little over 4/ΔT , then the received signal contains only the fundamental frequency 4/Δ*T*, i.e. it is a sine wave of frequency 4/Δ*T* Hz.
- The receiver can still decide correctly that the transmitted message was 01010101, so that the minimum bandwidth required for transmission of the square wave is 0 to 4/Δ*T*, i.e. 0 to 4fs Hz (since fs = 1/ΔT).
- Since this square wave has the highest frequency components of all possible pulse patterns, then the minimum bandwidth required for transmission of the PCM signal is 0 to 4fs. Thus in the general case of a single signal, sampled at fs and encoded into an *n*-bit code, we have:

Minimum PCM bandwidth for a single signal

$$\text{PCM bandwidth} = 0 \text{ to } \tfrac{1}{2} n f_S$$

Instrumentation

- For m multiplexed signals, each sampled at fs = 1/ΔT, the time in the HOLD state is ΔT/m.
- This means that n bits of information must be transmitted during time ΔT/m; i.e. the width of each bit of information is now ΔT/mn.
- The corresponding PCM bandwidth in this case is:

Minimum PCM bandwidth for m *multiplexed signals*

$$\text{PCM bandwidth} = 0 \text{ to } \tfrac{1}{2} m n f_S$$

Instrumentation

- We see that a single general expression for minimum PCM bandwidth is:

$$\text{PCM bandwidth} = 0 \text{ to } \tfrac{1}{2}R$$

- Thus a PCM signal, derived from 16 multiplexed signals, each sampled once per second and encoded into 12 bits, has a bit rate of $16 \times 12 = 192$ bauds and requires a transmission link with a minimum bandwidth of 0 to 96 Hz.